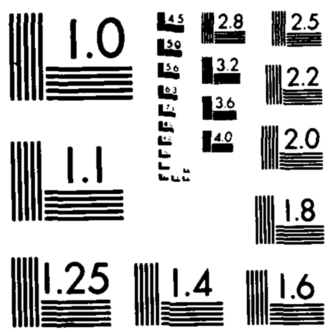
AD-A173 262 EQUIPMENT ARRANGEMENT USING INTERACTIVE COMPUTER GRAPHICS ON PERSONAL COMPUTERS(U) MASSACHUSETTS INST OF TECH CAMBRIDGE DEPT OF OCEAN ENGINEERING M J MEYERS 1/1

UNCLASSIFIED JUN 86 N00228-85-G-3262 F/G 9/2 NL

END
12 86
DTIC



ICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

DEPARTMENT OF OCEAN ENGINEERING

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

CAMBRIDGE, MASSACHUSETTS 02139

EQUIPMENT ARRANGEMENT USING
INTERACTIVE COMPUTER GRAPHICS
ON PERSONAL COMPUTERS

by

Michael J. Meyers

OCEAN ENGINEERING - COURSE XIII-A

JUNE 1986

1

EQUIPMENT ARRANGEMENT USING INTERACTIVE COMPUTER GRAPHICS ON PERSONAL COMPUTERS

by

Michael J. Meyers

B.S., Michigan Technological University (1974)

SUBMMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS OF THE
DEGREE OF

MASTER OF SCIENCE

IN NAVAL ARCHITECTURE AND MARINE ENGINEERING

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 1986

Signature of Author Michael J. Meyers
Department of Ocean Engineering, May 9 , 1986

Certified by C. Graham
Professor C. Graham
Thesis Supervisor

Accepted by A. Douglas Carmichael
Professor A. Douglas Carmichael, Chairman
Departmental Graduate Committee
Department of Ocean Engineering

EQUIPMENT ARRANGEMENT USING

INTERACTIVE COMPUTER GRAPHICS

ON PERSONAL COMPUTERS

by

MICHAEL J. MEYERS

Submitted to the Department of Ocean Engineering
on May 9, 1986 in partial fulfillment of the
requirements for the Degree of Master of Science in
Naval Architecture and Marine Engineering

NAVAl POSTGRADUATE SCHOOl
MONTEREY, CA 93943-5000
N00228-85-G-3262

ABSTRACT

A program was developed to facilitate the arrangement of equipment in a ship compartment utilizing interactive computer generated graphics from a parts/equipment library. The program is designed to send equipment arrangement locations via file transfer to a ship's data base.

The program was written in User Programming Language (UPL) computer language for an enhanced IBM AT personal computer running the ComputerVision Personal Designer MicroCADDS software.

Thesis Supervisor: Capt. C. Graham

Title: Professor of Ocean Engineering

TABLE OF CONTENTS

LIST OF FIGURES

Chapter 1

INTRODUCTION

1.1 General Background

The US Navy has been developing computer supported design capabilities in all major ship design functional areas for many years. The principle shortfall of these design capabilities has been thier lack of intergration among functional areas. The Navy's present effort is to produce an intergrated computer supported design process utilizing a common database system (Navy Intergrated Database or IDB). Part of this effort will be to develop the capability to produce equipment drawings utilizing Computer Aided Design (CAD) systems and to intergrate the internal compartment arrangement data into the Navy's IDB.

In 1984 the Naval Sea Systems Command of the United States Navy sponsored a Massachusetts Institute of Technology Sea Grant research project [1] in this area utilizing minicomputers with dedicated Computer Aided Design (CAD) hardware and software systems. The conclusions from the Sea Grant project form the underlying foundations of this project.

1.2 Problem Definition

The problem was to develop the capability to produce compartment internal equipment arrangement drawings on a desktop personal computer using the ComputerVision Personal Designer™ MicroCADDS computer software, and to create a program capable of transfering equipment arrangement location data to and from the ship's integrated database. A comparison of the capabilities of the different systems used in creating equipment arrangements and in interfacing with the IDB (ie. dedicated minicomputer CAD system vs. personal computer based CAD software) would point out the advantages and disadvantages of the system.

Chapter 2

METHODOLOGY

2.1 Project Guidlines

The direction and scope of this project comes from the trade-offs, conclusions, and recommendations of the MIT Sea Grant Project NA84AA-D-00046 R/T-28 [1]. The major tradeoffs and conclusions derived from equipment constraints, problem definition, and end user interaction are:

(1) Whether to develop a program that has universal application or one that is machine specific?[2] The basic problem being that univerality requires much greater detail which means a much greater volume of data to be transfered. The program developed will be machine (equipment) specific.

(2) Should the program use the IGES (Initial Graphics Exchange Specification) or the ASCII format? The IGES format is designed to transfer graphic data and detail between normally incompatable machines, therefore lending a quasi-universality to the program. The major dificulties with the IGES format are that it takes an enormous amount of time to transfer all the graphic data, and it requires machine specific extractors and processor

be written for both the sending and receiving machines. If the process is to be machine specific then the ASCII format will be adopted because if it's ease of use, and it's speed in transfer [3].

(3) What <u>level of detail</u> is required? There are several reasons that lead to the decision to send only the absolute minimum of information between computers. The most prominent ones are to reduce the amount of time involved in data transfer, and to ascertain the accuracy of the data sent [4]. The program was developed to keep the level of detail down to the minimum required. Therefore the graphic geometries (neither compartment, hull, nor equipment) are not sent back to the IDB once they are resident in the CAD computer [5].

As can readily be seen all of these trade-offs are interrelated and seem to have evolved as the project developed.

An overriding concern was that the program developed have a high degree of user friendliness. This should translate into requiring a minimal numberof inputs from the individual users and only a nominal understanding of the internal mechanism of the program itself.

The above stated trade-off results and conclusions from MIT Sea Grant Report 86-3TN [1] were used as guidelines for this poject to use in program development.

2.2 Basic Approach

The approach used was the one that would minimize the amount of data transfered between the arranging computer and the one on which the database would formally reside. A parts library was created to contain the full 3-D image of each piece of equipment. This image was stored as a figure. The figure origin was placed at the lower left forward corner of a piece of equipment unless the nature of that equipment dictated otherwise. The figures were inserted into the compartment drawing and arranged as desired. All arrangements are done in view 1 (Top View) to preclude problems with the depth (Z coordinate) setting. When the arrangement faze is completed the location information of each equipment is extracted from the compartment drawing and stored in a file. This file will then be either reprocessed or sent in toto to the IDB so that the location information can be removed there. The reverse process would be to take the returning file from the IDB, read the incoming data, and then insert the figures at the locations specified.

2.3 Equipment

The hardware utilized was an IBM Personal Computer AT with an installed 20 Megabyte Hard disk drive (c:), 1.2 Megabyte High capacity floppy diskette drive (a:), 512 Kilobyte onboard RAM (Random Access Memory), 128 Kilobyte RAM (Random Access Memory) on an AST Research INC. Advantage card with serial/parallel I/O (Input/Output) adaptors (comm2/lpt2), Tecmar Graphics Master color screen adaptor card, an IBM serial/parallel I/O adaptor (comm1/lpt1), and an 8087 math coprocessor. The attached peripherals include an Epson LQ-1000 printer (lpt1), Sweet-P plotter (comm1), Prometheus ProModem1200 modem (comm2), and a Hitachi HDG-1111B Digitizing Tablet (upgraded to 12x12 work area)(comm1).

The computer aided design (CAD) software utilized was developed by ComputerVision. The software included were ComputerVision's microCADDS Personal Designer [6], Utilities [7], Personal Systems Communication [8], and the User Programming Language (UPL) [9]. The other major "software" used was IBM's Disk Operating System (DOS) version 3.1 and the IBM Line Editor (EDLIN) [10].

Chapter 3

RESULTS

3.1 Basic Program Description

The program is designed to be run from within the microCADDS Geometric Construction and Detailing (GCD) processor. The user would be within the computer aided design (CAD) environment, inside a compartment drawing, and then utilizes the TRANSFER program in either a receiving or preparing mode to manipulate the equipment data.

The program (TRANSFER.UCD) is composed of four main sections. These are the GROUP variable declaration and three PROCESSes: the Program Executive, the subroutine "OUTPUT", and the subroutine "INPUT". The Program Executive is the main process. It directs the user to the appropriate subroutine for action. The subroutine "INPUT" receives an equipment data file from the IDB, reads it, and then either inserts figures into a compartment drawing or erases figures from a compartment drawing. The subroutine "OUTPUT" prepares an equipment data file that will be sent as output to the IDB.

3.2 General Output

The output of this program was a data file that may be sent either via modem, tape, or floppy diskette to the mainframe computer that contains the IDB. The data file produced is a sequential mode formated human readable text-type file. The data file is intended to be writen to, read from, and modified from either the arranging computer or the one on which the IDB formally resides.

There exists another option for file transfer and manipulation. A part of the ComputerVision Personal Designer software package is a program called Personal CADDS Connect (PCC) [8] which will allow the user on a personal computer to connect to one of the dedicated minicomputers running the CADDS 4X software (described in the MITSG Report 86-3TN [1]). The user would be able to transfer the data file to or from the larger CAD system or be able to operate the personal computer as a CADDS 4X workstation with access to all the graphics library parts (NFIGs) stored on the other system.

3.3 RESULTS AND CONCLUSIONS

This project resulted in the capability to create internal compartment equipment arrangements, and produced a program to manipulate those equipment arrangement locations. The compartment internal arrangements were made by inserting and arranging previously created library part equipment figures into the compartment. The program was created using the User Programing Language (UPL). It's function was to allow equipment loctation data to be sent to or recieved from the Navy's Integrated Data Base (IDB). The complete description of these functions is detailed in the Appendices of this report. The ultimate result of this project was that all this was done on a desktop computer system (detailed in Section 2.3).

That this could be achieved on a desktop system vice a dedicated system points out the versitility and utility of these systems. The ComputerVision Personal Designer™ microCADDS system is an extremely powerful CAD system which has about 85 percent of the command language of the larger more costly dedicated CADDS4X system (detailed in referance [1]). The system is easy to use and is relatively user friendly. The programming language UPL has a good logical structure, a fast compile time, and adequate documentation. The sole problem with the UPL and microCADDS system is the

long tedious debugging cycle that requires a seperate compile, boot of the GCD processor, and then test of the program code. The major drawback of the desktop CAD system itself is that it has available less than one tenth the available core memory of the dedicated systems. This is being overcome by the addition of advanced storage devices (Bernoulli boxes, add-on hard disk drives, bubble memory cards, etc.) or by tying into a larger system (see Section 3.2) via modem or cable in order to access stored parts libraries and databases.

In comparing the program written here and the ones designed for the larger system [1] the basic differences are minor and cosmetic at most. The desktop program is a single program performing all the functions of the two programs written for the CADDS4x system. It does require more user input (keystrokes) and the insert function should be done in the top view to avoid an orientation problem. These are not necessarily limitations however they are the major differances between the two programs.

It is recommended that equipment should not be arranged or rearranged from the IDB end. This is because the spacial relationships and interferances are extremely difficult if not impossible to visualize without graphic 3-D representation. The most that should be done from the

IDB end would be to eliminate (erase) unwanted pieces of equipment.

Appendix I

USER'S MANUAL

A. Detailed Program Description

The program (TRANSFER.UCD) described in this appendix performs the tasks of extracting information from the graphics database of the arranging computer and writing it to a formatted output file, or by taking a similarly formatted file as input, reading it and causing graphic fiqures to be inserted into an empty ship's compartment or to be erased from an existing prearranged ship's compartment. This program was written in User Programming Language (UPL) running under the ComputerVision Personal Designer microCADDS system. While this appendix does not require intimate knowledge of the microCADDS system it does require basic knowledge of computer aided design (CAD) concepts and environment.

In this program each output data file created contains information about a single ship's compartment referanced to the ship's origin or the compartment's origin as initially defined on the computer's CAD system. This data file is a sequential mode formated human readable text-type file. The basic line of data in the file contains information

about each piece of equipment in the compartment whose geometry has been previously defined as a figure in the parts library. In a compartment containing library defined parts and non-library entities only those pieces of equipment that are library parts can be sent or received.

The program (TRANSFER.UCD) is composed of four main sections. They are the GROUP variable declaration, and three PROCESSes: the Program Executive, the subroutine "Output", and the subroutine "Input". The GROUP section declares global variables (usable and accessable by all routines and processes, and having the same names and arguments), sets string lengths, and array dimensions. The Program Executive performs the basic function of traffic director by offering the user a menu of choices and then sending the user to the appropriate subroutine that performs the desired function. The subroutine "Output" prepares an output file of equipment information to be sent to the IDB. The subroutine "Input" receives an input file, allows the user to insert or erase equipment figures from an existing compartment.

The subroutine "Output" begins by asking the user to input the compartment name and the number of pieces of equipment that will have information prepared for. The subroutine then uses two digitizes per equipment figure to

obtain the coordinate information of the equipment's origin (FIGURE_ORIGIN(I)) and the Master Index Block number (MIB(I)) of the figure's location in the graphics database by the use of some intrinsic UPL functions. The user is then prompted to input the equipment's library part (figure) name (GENERIC_NAME(I)) and that equipment's unique descriptor (UNIQUE_NAME(I)). It iterates as many times as necessary to get all the pieces of equipment that the user has said will have information prepared about. Next the output file is written. Finally the user is prompted as to whether to continue with this part of the program or to return to the Program Executive.

The subroutine "Input" consists of two distinct parts: the insertion mode and the erasure mode. The subroutine starts by asking the user for the compartment name (which is used as the file name when the extension ".DAT" is added to it) and the number of pieces of equipment contained in the file. The subroutine then reads the file and loads the arrays with the file's data. At this point the user is asked if equipment insertion or erasure is desired. These two operations are mutually exclusive and should have separate data files prepared for each operation. In insertion mode the subroutine uses the Geometric Construction and Detailing (GCD) resident master program to "INSERT" the figures at the appropriate coordinate location

within the compartment. This mode also "INSERT"s (TEXT) the unique equipment part name (UNIQUE_NAME(I)) at the figure's origin. The erasure mode is a rapid wipe of the graphics database Master Index Block numbers (MIB(I)) read off of the input file. Finally this subroutine also prompts the user as to whether to continue with this part of the program or to return to the Program Executive.

B. Preparation Steps

Before the TRANSFER.UCD program can be run a compartment, and the equipment figures must be prepared. The compartment size and shape should be imported from the Approved version of the ship that would reside in the IDB. This compartment would then be refered to as the "part" by the microCADDS system and would be given the extension ".DRW" to indicate that it is a separate and distinct drawing. If the entire ship is brought over and made resident within the microCADDS system then the arranger would have to pick out the individual compartment within the system in order to perform the arranging.

To prepare the equipment figures for insertion into the compartment it is first necessary to incorparate them into the equipment parts library. Each piece of equipment is first created as a part. The opening inquery in the microCADDS system prompts for the part name. The response should be the generic equipment name that will refer to the geometry of this particular piece of equipment. The equipment's geometry is stored under a generic name so that the visual graphic representation can be called or inserted many times while refering to a single figure in the parts library. A point should be inserted at the figure's origin so that the figure's orientation will be clear through out

the creation of the part. This figure origin should be placed at the lower left forward corner of the specific equipment unless the nature of that equipment would dictate otherwise. For example, a chair or desk would have thier origin in the lower left forward corner (see figure 1 or 2), but an overhead fixture might have its origin in the upper center face of the equipment. Utilizing the modeling capabilities of the microCADDS system the equipment geometry should then be created and stored in the parts library. If the manufacturers of each piece of equipment used were to submit IGES compatible FIGUREs on magtape when providing the equipment [11] this preparation step could be eliminated. It is also possible to insert the generic equipment name into the FIGURE so that it will show up in every instance of the equipment in the compartment.

Once the parts library has been created and the compartment size and shape obtained, the compartment can be arranged and rearranged easily.

C. Sample Example Session

In this example we will go through the sequence of creating some new library parts, their insertion and arrangement in a compartment, and the use of the TRANSFER.UCD program. It is assumed that the reader understands the microCADDS Geometric Construction and Detailing (GCD) command structure [6] and the general use of CAD modeling concepts.

After having booted up the system, proceed to the Geometric Construction and Detailing (GCD) program to do the actual modeling. For this example the terminal screen prompt will be displayed as BOLD LETTERS and the user input will be displayed as ***BOLD ITALIC LETTERS***:

IBM TERMINAL DEFINITION

PART NAME: >>

This is the GCD prompt for the user to input the name of the part that will be created.

CHAIR

NEW PART

>> ***INSERT POINT :*** **dig** ***X 0 Y 0 Z 0***

This causes a point to be inserted at the origin of the drawing. This origin will become the FIGURE_ORIGIN which will be the point and orientation used when the part is

inserted into the compartment drawing as a figure. After this has been done the user may then perform the modeling steps required to complete the part drawing.

>> *ZOOM ALL*

>> *EXIT*

Filing Name: *CHAIR*

Save <Y, N, ^C>: *Y*

Replace Existing File <Y, N, ^C>: *Y*

Pack the Database ? <Y, N, ^C>: *Y*

In order to store the part as a figure a ZOOM ALL command must be issued followed by the appropriate FILE or EXIT command (see figures 1, 2, and 3). Next a compartment should be prepared or imported and then the figures can be inserted, arranged, or rearranged (see figure 4).

PART NAME: *COMPT1*

OLD PART DB VERSION 2

>> *RESTORE VIEW 7< 1*

>> *INSERT FIGURE* file = *CHAIR* *:*dig *d*

>> *MOVE* *:*ent *d*,dig from *ORG* *d*,to *d*

As many figures can be created and stored in the parts library as is necessary (and as storage space will allow). Once the parts library has been created and the compartment arranged the TRANSFER program may be run. The program is run from within the compartment drawing that will be affected.

>> *RUN TRANSFER*

MAIN MENU

A) PREPARE AN EQUIPMENT INFORMATION FILE

B) RECEIVE AN EQUIPMENT INFORMATION FILE

C) EXIT THIS PROGRAM

CHOOSE AN OPTION: A , B , or C ? *A*

INPUT THE COMPARTMENT NAME + .DAT: *COMPT1.DAT*

ENTER THE NUMBER OF PIECES OF EQUIPMENT. *4.*

From this point on when the user has entered all the pertinent data (and not a space extra) a period "." will terminate the entry but will not be read as part of the entry.

DIGITIZE THE DESIRED PIECE OF EQUIPMENT *d*

INPUT THE EQUIPMENT GENERIC NAME: *CHAIR.*

INPUT THE EQUIPMENT UNIQUE NAME: *C1.*

The user will repeat this cycle as many times as needed until all the pieces of equipment have been picked up. The output file (COMPT1.DAT) is automatically written (see Section D of this appendix).

ARE YOU DONE WITH PART A OF THE PROGRAM? *Y*

ARE YOU THROUGH WITH THIS PROGRAM? *N*

At this point the user could have answered *Yes* and the user would have been returned to the microCADDS environment. For the purposes of this USERS MANUAL the example will continue on.

MAIN MENU

A) PREPARE AN EQUIPMENT INFORMATION FILE

B) RECEIVE AN EQUIPMENT INFORMATION FILE

C) EXIT THIS PROGRAM

CHOOSE AN OPTION: A , B , or C ? *B*

INPUT THE COMPARTMENT NAME + .DAT: *COMPT1.DAT*

ENTER THE NUMBER OF PIECES OF EQUIPMENT. *4.*

Here the program automatically reads the input file (COMPT1.DAT) before proceeding to the next prompt.

DO YOU WISH TO I)NSERT OR E)RASE EQUIPMENT? *I*

The program manipulates the data from the input file, transfers control to the GCD processor, and inserts figures and text at the appropriate locations.

ARE YOU DONE WITH PART B OF THE PROGRAM? *N*

Again for the purposes of this USERS MANUAL the example will continue on.

INPUT THE COMPARTMENT NAME + .DAT: *COMPT1.DAT*

ENTER THE NUMBER OF PIECES OF EQUIPMENT. *4.*

Here the program automatically reads the input file (COMPT1.DAT) before proceeding to the next prompt.

DO YOU WISH TO I)NSERT OR E)RASE EQUIPMENT? *E*

The program manipulates the data from the input file, transfers control to the GCD processor, and erases figures from the appropriate locations.

ERASING FILER FA

ERASING DESK D1

ERASING CHAIR C1

ERASING FILER FB

ENTITIES ERASED

ARE YOU DONE WITH PART B OF THE PROGRAM? *Y*

ARE YOU THROUGH WITH THIS PROGRAM? *Y*

The user is now returned to the microCADDS environment, and this example is completed.

D. Detailed Output

The data file produced by the TRANSFER.UCD program is a sequential mode formatted human readable text-type file. It is sequential mode in that each variable line follows the previous line in sequential order until the end of the file is reached. It is formatted in that within each variable line there is a specific location, variable type, and length for each variable. The human readable text-type refers to its manner of being written in ASCII characters rather than binary machine code. The write statement for the file is:

```
     "LOOP I = 1 TO J
      WRITE F1, MIB(I):-10, GENERIC_NAME(I):10, \
UNIQUE_NAME(I):10, FIGURE_ORIGIN(I):30:3
      END_LOOP"
```

This write statement produces "J" variable lines with four variables per line. These variables are:

MIB(I):-10 - Master Index Block integer array element which has a field width of 10 characters and is left justified within the field

GENERIC_NAME(I):10 - array element containing the string name of the generic equipment figure in the parts library, the element has a field width of 10 characters and

is right justified within the field

UNIQUE_NAME(I):10 - array element containing the string name that is the unique name of the particular instance of that piece of equipment, the element has a field width of 10 characters and is right justified within the field

FIGURE_ORIGIN(I):30:3 - array element containing the coordinate location of a particular instance of a figure's origin, the element has a field width of 25 with 4 significant decimals retained, and is right justified within the field.

The following is the example file COMPT1.DAT; given in its listed format of:

MIB	GENERIC_NAME	UNIQUE_NAME	FIGURE_ORIGIN
294	FILER	FA	[10.963,89.444,.000]
136	DESK	D1	[91.636,31.436,.000]
236	CHAIR	C1	[65.121,34.624,.000]
293	FILER	FB	[179.494,31.490,.000]

Appendix II

LISTINGS

A. Variable Listings

The following is a list of the variable names used in the program. Included in this listing will be name descriptors and variable types used.

Variable	Type	Name Descriptor
NEND	Integer	Number of ends
NENT	Integer	Number of entities
MIB(I)	Integer	Master Index Block array
ANS	String	Answer to a query
GENERIC_NAME(I)	String	Name of generic figure
UNIQUE_NAME(I)	String	Unique equipment name
COMPT_NAME	String	Compartment name
FIGURE_ORIGIN(I)	Coordinate	The figure's origin
I	Integer	Counting marker
J	Integer	Counting marker
INT	Integer	Entity type number
NBREAD	Integer	Number of bytes read from database subrecord
IERR	Integer	Operation error code

Variable	Type	Name Descriptor
TF	String	Text File type of subrecord
FL	File	File name variable
C(6)	Coordinate	Data locations within the subrecord database
BUF	String	Data input from subrecord data base
X	Real	X coordinate
Y	Real	Y coordinate
Z	Real	Z coordinate

B. Program Listing

The following is the program listing:

```
---------------------------------------------------------------

------------------------------DECLARE VARIABLES

GROUP

INTEGER NEND

INTEGER NENT

INTEGER INT

INTEGER NBREAD

INTEGER IERR

INTEGER MIB(20)

STRING  ANS:1

STRING  TF:2

STRING  GENERIC_NAME(20):10

STRING  UNIQUE_NAME(20):10

STRING  COMPT_NAME:10

COORD   FIGURE_ORIGIN(20)

FILE    FL

COORD   C(6)

STRING  BUF:72 @ C

REAL    X

REAL    Y

REAL    Z

END GROUP

------------------------------SUBROUTINE "OUTPUT"
```

```
PROC OUTPUT
INTEGER I,J
NEND=0
NENT=0
TF = 'TF'
---------------
POINT1:
PRINT
ACCEPT COMPT_NAME \
       PROMPT('INPUT THE COMPARTMENT NAME + .DAT') \
       NEWLINE
OPEN F1 COMPT_NAME
ACCEPT J \
       PROMPT('ENTER THE NUMBER OF PIECES OF EQUIPMENT.') \
       LAST('.') NEWLINE
---------------
POINT2:
LOOP I = 1 TO J
POINTA:
PRINT 'DIGITIZE THE DESIRED PIECE OF EQUIPMENT: '
PRINT
GETENT(1,NENT,MIB(I),IEND(I))
VERIFY ENTTYP(INT), ENTID(MIB(I))
IF INT = 11 THEN
     GETSR(MIB(I),TF,1,62,NBREAD,BUF,IERR)
     FIGURE_ORIGIN(I) = C(5)
```

```
        ACCEPT GENERIC_NAME(I) LAST('.') \
        PROMPT('INPUT THE EQUIPMENT GENERIC NAME:') \
        NEWLINE
        ACCEPT UNIQUE_NAME(I) LAST('.')  \
        PROMPT('INPUT THE EQUIPMENT UNIQUE PART NAME:') \
        NEWLINE
    ELSE
        GOTO POINTA
    ENDIF
    END_LOOP
    SEND 'REPAINT'
    ---------------
    POINT3:
    LOOP I = 1 TO J
    WRITE FL, MIB(I):-10,GENERIC_NAME(I):10,UNIQUE_NAME(I):10,\
              FIQURE_ORIGIN(I):30:3
    END_LOOP
    CLOSE FL
    ---------------
    POINT4:
    ACCEPT ANS\
          PROMPT('ARE YOU DONE WITH PART A OF THE PROGRAM?')\
          NEWLINE
    IF ANS= 'Y' OR ANS= 'y' THEN
        GOTO POINT5
    ELSE IF ANS= 'N' OR ANS= 'n' THEN
```

```
    GOTO POINT1
ELSE
    PRINT ANS,' IS NOT AN OPTION. CHOOSE AGAIN. '
    PRINT
    GOTO POINT4
ENDIF
---------------
POINT5:
RETURN
END PROC
--------------------------------SUBROUTINE "INPUT"
PROC INPUT
INTEGER I,J
NEND=0
NENT=0
---------------
POINT1:
ACCEPT COMPT_NAME \
       PROMPT('INPUT THE COMPARTMENT NAME + .DAT') \
       NEWLINE
OPEN FL COMPT_NAME
ACCEPT J \
       PROMPT('ENTER THE NUMBER OF PIECES OF EQUIPMENT.')\
       LAST('.') NEWLINE
---------------
POINT2:
```

```
LOOP I = 1 TO J
READ FL, MIB(I):-10,GENERIC_NAME(I):10,UNIQUE_NAME(I):10,\
          FIGURE_ORIGIN(I):30
END_LOOP
CLOSE FL
---------------
POINT3:
ACCEPT ANS \
     PROMPT('DO YOU WISH TO I)SERT OR E)RASE EQUIPMENT?')\
     NEWLINE
IF ANS='I' OR ANS='i' THEN
     GOTO POINT4
ELSE IF ANS= 'E' OR ANS= 'e' THEN
     GOTO POINT5
ELSE
     PRINT ANS,' IS NOT AN OPTION. CHOOSE AGAIN.'
     PRINT
     GOTO POINT3
ENDIF
---------------
POINT4:
LOOP I = 1 TO J
GENERIC_NAME(I) = RMV_CHR(GENERIC_NAME(I),'#9 #13')
UNIQUE_NAME(I) = RMV_CHR(UNIQUE_NAME(I),'#9 #13')
SEND 'INS FIG ',
SEND GENERIC_NAME(I),' :',
```

```
SEND ' X ',FIGURE_ORIGIN(I).X,
SEND ' Y ',FIGURE_ORIGIN(I).Y,
SEND ' Z ',FIGURE_ORIGIN(I).Z
INSERT TEXT TXT(UNIQUE_NAME(I)) , ORG(FIGURE_ORIGIN(I)) \
                , HGT(4.0) , WDT(4.0) , LNSP(3.0)
END_LOOP
SEND 'REPAINT'
GOTO POINT6
---------------
POINT5:
LOOP I = 1 TO J
ERASE ENT_ID(MIB(I))
PRINT 'ERASING ',GENERIC_NAME(I),' ',UNIQUE_NAME(I)
END_LOOP
SEND 'REPAINT'
PRINT 'ENTITIES ERASED'
---------------
POINT6:
ACCEPT ANS \
       PROMPT('ARE YOU DONE WITH PART B OF THE PROGRAM?')\
       NEWLINE
IF ANS= 'Y' OR ANS= 'y' THEN
     GOTO POINT7
ELSE IF ANS= 'N' OR ANS= 'n' THEN
     GOTO POINT1
ELSE
```

```
     PRINT ANS,' IS NOT AN OPTION. CHOOSE AGAIN.'
     PRINT
     GOTO POINT6
ENDIF
---------------
POINT7:
RETURN
END PROC
--------------------------------PROGRAM EXECUTIVE
PROC MAIN
START:
SEND
SEND 'SET SCROLL ','6'
---------------MAIN MENU
DISPLAY
               MAIN MENU
     A) PREPARE AN EQUIPMENT INFORMATION FILE
     B) RECEIVE AN EQUIPMENT INFORMATION FILE
     C) EXIT THIS PROGRAM
          CHOOSE AN OPTION: A , B , or C ? $
ACCEPT ANS NEWLINE
IF ANS= 'A' OR ANS= 'a' THEN
     OUTPUT
ELSE IF ANS= 'B' OR ANS= 'b' THEN
     INPUT
ELSE
```

```
     PRINT ANS,' IS NOT AN OPTION. CHOOSE AGAIN.'
     PRINT
     GOTO START
ENDIF
--------------CLOSING MENU
DONE:
ACCEPT ANS PROMPT(' ARE YOU THROUGH WITH THIS PROGRAM? ')\
       NEWLINE
IF ANS= 'Y' OR ANS= 'y' THEN
     GOTO ENDALL
ELSE IF ANS= 'N' OR ANS= 'n' THEN
     GOTO START
ELSE
     PRINT ANS,' IS NOT AN OPTION. CHOOSE AGAIN.'
     PRINT
     GOTO DONE
ENDIF
--------------END OF PROGRAM
ENDALL:
END PROC
```

LIST OF REFERANCES

[1] C.Chryssostomidis,C.Graham,M.Meyers,P.V.Prakash; "EQUIPMENT ARRANGEMENT USING INTERACTIVE COMPUTER GRAPHICS"; MIT SEA GRANT Project NA84AA-D-00046 R/T-28 Report No. MITSG 86-3TN; APRIL 1986

[2] C.Chryssostomidis,C.Graham,M.Meyers,P.V.Prakash; "EQUIPMENT ARRANGEMENT USING INTERACTIVE COMPUTER GRAPHICS"; MIT SEA GRANT Project NA84AA-D-00046 R/T-28 Report No. MITSG 86-3TN; APRIL 1986; Page 4

[3] C.Chryssostomidis,C.Graham,M.Meyers,P.V.Prakash; "EQUIPMENT ARRANGEMENT USING INTERACTIVE COMPUTER GRAPHICS"; MIT SEA GRANT Project NA84AA-D-00046 R/T-28 Report No. MITSG 86-3TN; APRIL 1986; Page 5

[4] C.Chryssostomidis,C.Graham,M.Meyers,P.V.Prakash; "EQUIPMENT ARRANGEMENT USING INTERACTIVE COMPUTER GRAPHICS"; MIT SEA GRANT Project NA84AA-D-00046 R/T-28 Report No. MITSG 86-3TN; APRIL 1986; Page 5

[5] C.Chryssostomidis,C.Graham,M.Meyers,P.V.Prakash; "EQUIPMENT ARRANGEMENT USING INTERACTIVE COMPUTER GRAPHICS"; MIT SEA GRANT Project NA84AA-D-00046 R/T-28 Report No. MITSG 86-3TN; APRIL 1986; Page 6

[6] "microCADDS Geometric Construction and Detailing Referance"; Revision 2.0; ComputerVision Corporation October 1985

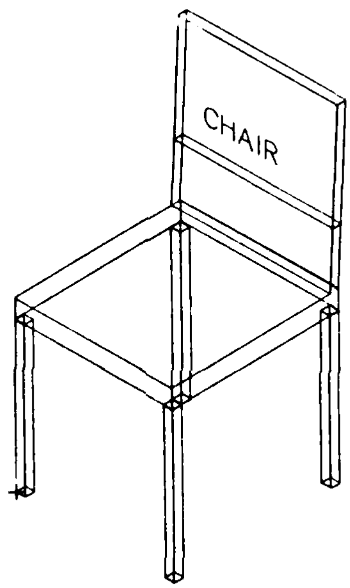
[7] "microCADDS Geometric Construction and Detailing Referance"; Revision 2.0; ComputerVision Corporation October 1985; Appendix A

[8] "Persnoal Systems Communication Referance"; Revision 2.0; ComputerVision Corporation; November 1985

[9] "User Programing Language Referance"; Revision 2.0; ComputerVision Corporation; October 1985

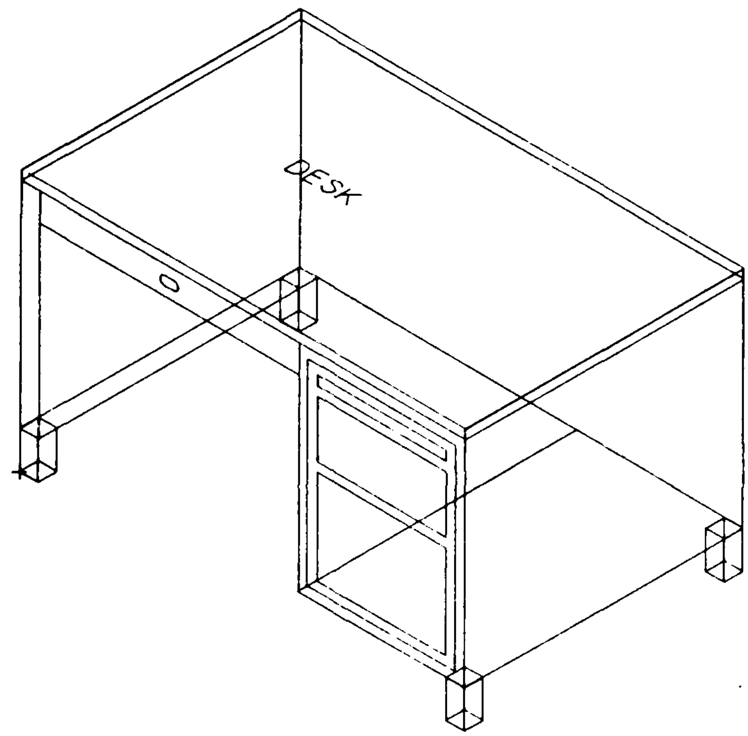
[10] Disk Operating System (DOS) Version 3.10 International Buisness Machines Corporation (IBM) 1984,1985

[11] C.Chryssostomidis,C.Graham,M.Meyers,P.V.Prakash; "EQUIPMENT ARRANGEMENT USING INTERACTIVE COMPUTER GRAPHICS"; MIT SEA GRANT Project NA84AA-D-00046 R/T-28 Report No. MITSG 86-3TN; APRIL 1986; Page 9



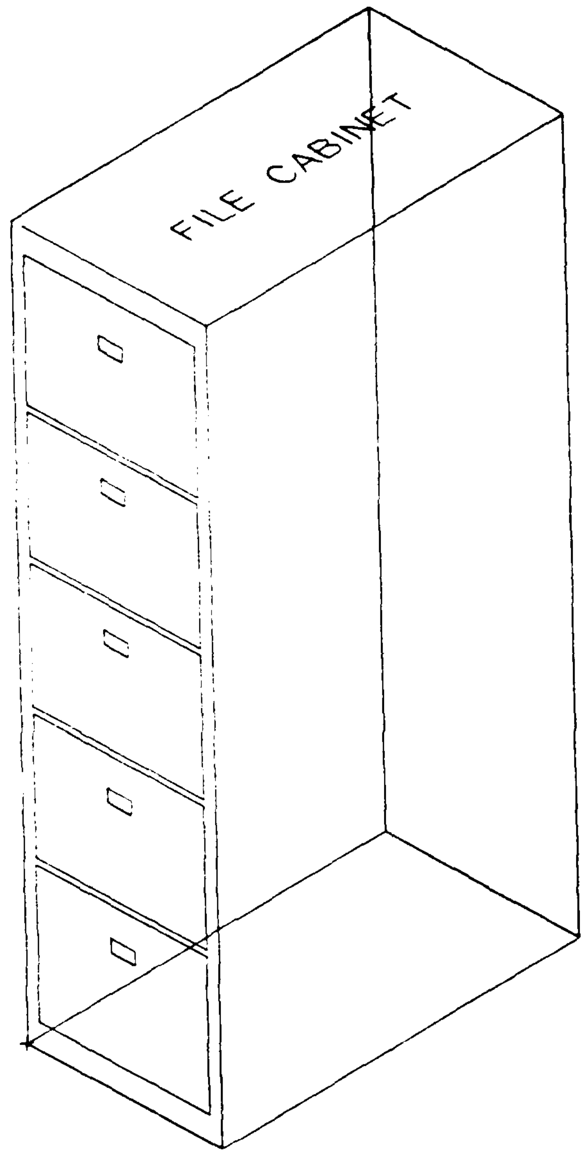
CHAIR

FIGURE 1



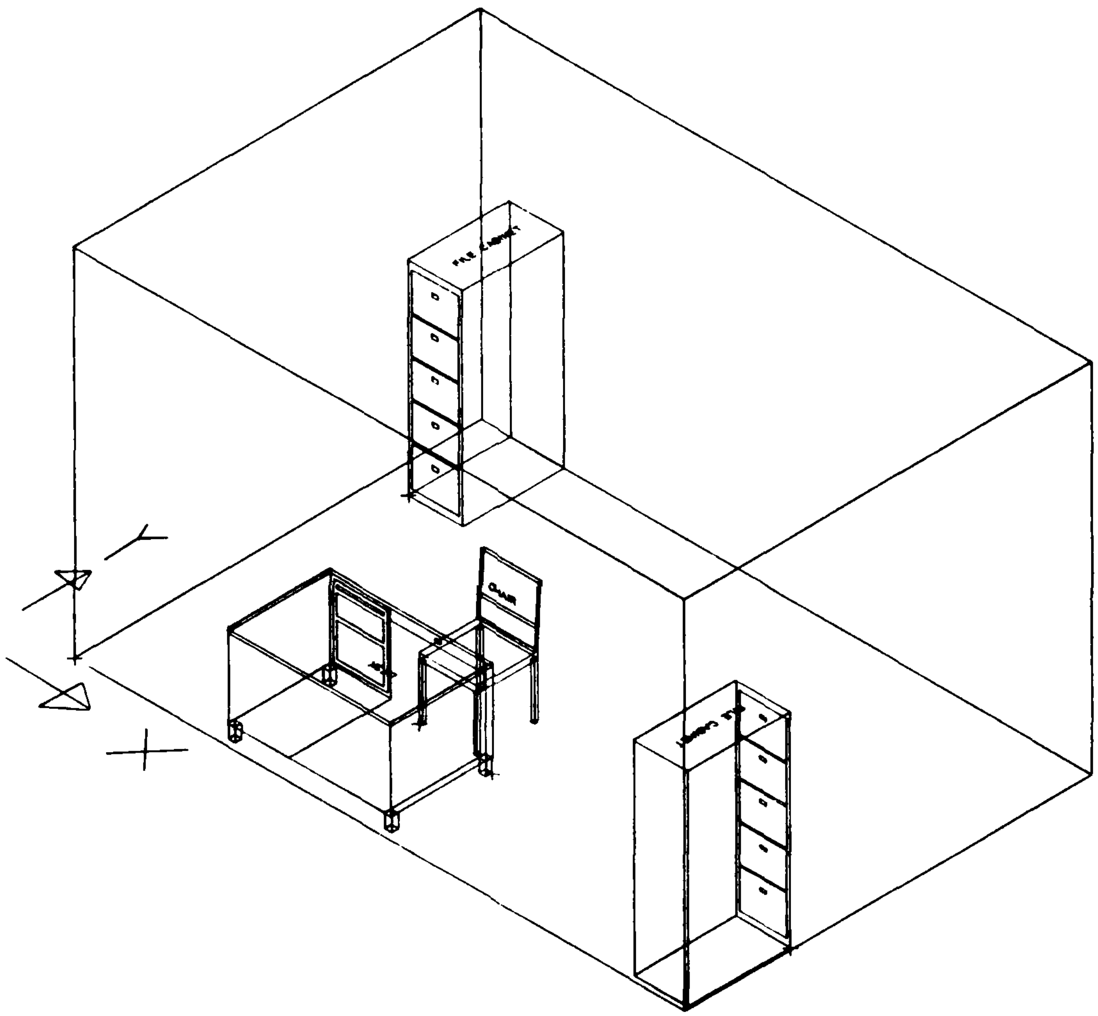
DESK

FIGURE 2



FILER

FIGURE 3



ARRANGED COMPARTMENT

FIGURE 4

END

12-86

DTIC